52. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

PROCCEDINGS | 10 - 13 September 2007

FACULTY OF COMPUTER SCIENCE AND AUTOMATION

COMPUTER SCIENCE MEETS AUTOMATION

VOLUME I

Session 1 - Systems Engineering and Intelligent Systems

Session 2 - Advances in Control Theory and Control Engineering

Session 3 - Optimisation and Management of Complex Systems and Networked Systems

Session 4 - Intelligent Vehicles and Mobile Systems

Session 5 - Robotics and Motion Systems

TECHNISCHE UNIVERSITÄT
ILMENAU

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der deutschen Nationalbiografie; detaillierte bibliografische Daten sind im Internet über **http://dnb.ddb.de** abrufbar.

ISBN 978-3-939473-17-6

Impressum

Herausgeber:	Der Rektor der Technischen Universität Ilmenau Univ.-Prof. Dr. rer. nat. habil. Peter Scharff
Redaktion:	Referat Marketing und Studentische Angelegenheiten Kongressorganisation Andrea Schneider Tel.: +49 3677 69-2520 Fax: +49 3677 69-1743 e-mail: kongressorganisation@tu-ilmenau.de
Redaktionsschluss:	Juli 2007
Verlag:	Technische Universität Ilmenau/Universitätsbibliothek Universitätsverlag Ilmenau Postfach 10 05 65 98684 Ilmenau www.tu-ilmenau.de/universitaetsverlag
Herstellung und Auslieferung:	Verlagshaus Monsenstein und Vannerdat OHG Am Hawerkamp 31 48155 Münster www.mv-verlag.de
Layout Cover:	www.cey-x.de
Bezugsmöglichkeiten:	Universitätsbibliothek der TU Ilmenau Tel.: +49 3677 69-4615 Fax: +49 3677 69-4602

© Technische Universität Ilmenau (Thür.) 2007

Diese Publikationen und alle in ihr enthaltenen Beiträge und Abbildungen sind urheberrechtlich geschützt. Mit Ausnahme der gesetzlich zugelassenen Fälle ist eine Verwertung ohne Einwilligung der Redaktion strafbar.

Preface

Dear Participants,

Confronted with the ever-increasing complexity of technical processes and the growing demands on their efficiency, security and flexibility, the scientific world needs to establish new methods of engineering design and new methods of systems operation. The factors likely to affect the design of the smart systems of the future will doubtless include the following:

- As computational costs decrease, it will be possible to apply more complex algorithms, even in real time. These algorithms will take into account system nonlinearities or provide online optimisation of the system's performance.
- New fields of application will be addressed. Interest is now being expressed, beyond that in "classical" technical systems and processes, in environmental systems or medical and bioengineering applications.
- The boundaries between software and hardware design are being eroded. New design methods will include co-design of software and hardware and even of sensor and actuator components.
- Automation will not only replace human operators but will assist, support and supervise humans so that their work is safe and even more effective.
- Networked systems or swarms will be crucial, requiring improvement of the communication within them and study of how their behaviour can be made globally consistent.
- The issues of security and safety, not only during the operation of systems but also in the course of their design, will continue to increase in importance.

The title "Computer Science meets Automation", borne by the 52nd International Scientific Colloquium (IWK) at the Technische Universität Ilmenau, Germany, expresses the desire of scientists and engineers to rise to these challenges, cooperating closely on innovative methods in the two disciplines of computer science and automation.

The IWK has a long tradition going back as far as 1953. In the years before 1989, a major function of the colloquium was to bring together scientists from both sides of the Iron Curtain. Naturally, bonds were also deepened between the countries from the East. Today, the objective of the colloquium is still to bring researchers together. They come from the eastern and western member states of the European Union, and, indeed, from all over the world. All who wish to share their ideas on the points where "Computer Science meets Automation" are addressed by this colloquium at the Technische Universität Ilmenau.

All the University's Faculties have joined forces to ensure that nothing is left out. Control engineering, information science, cybernetics, communication technology and systems engineering – for all of these and their applications (ranging from biological systems to heavy engineering), the issues are being covered.

Together with all the organizers I should like to thank you for your contributions to the conference, ensuring, as they do, a most interesting colloquium programme of an interdisciplinary nature.
I am looking forward to an inspiring colloquium. It promises to be a fine platform for you to present your research, to address new concepts and to meet colleagues in Ilmenau.

Professor Peter Scharff
Rector, TU Ilmenau

Professor Christoph Ament
Head of Organisation

Table of Contents

C O N T E N T S

Paper submitted after copy deadline

2 Advances in Control Theory and Control Engineering

52nd Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
10 – 13 September 2007

M. Kaufmann / G. Bretthauer

Development and composition of control logic networks for distributed mechatronic systems in a heterogeneous architecture

INTRODUCTION

Information processing, plays an ever-increasing role in the functionality and quality of products in the field of mechanical engineering and becomes an inextricable part of the system function. However, program execution requires special hardware separated from the mechanical and electrical components. Typical hardware architectures in this domain are embedded systems (Micro controller, DSP board, micro processor, – with/without operating system), personal computer programmable logic controller (PLC) – and composed of several components: distributed systems with central supervisory control, distributed systems with distributed control, distributed autonomous cooperative systems. In many cases, a heterogeneous architecture consisting of diverse electronic units, software components, and communication interfaces is implemented (see Figure 1).

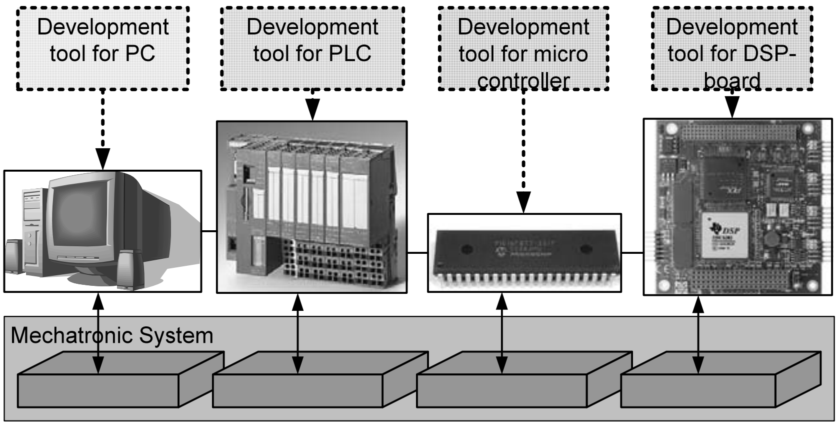

Figure 1: Mechatronic system with heterogeneous hardware and software architecture

Inside the different hardware units similar functional entities are executed – typically described by functional blocks. Among them are logical networks – consisting of logical operations, controllers – closed loop and open loop, various arithmetic operations, combinations of the aforementioned entities.

During the development of heterogeneous systems, different development tools using different standards, programming approaches and programming languages for basically similar functionalities are used. Developers make great efforts to verify functional consistency of these systems, i.e. to test interoperation of entities developed by different tools. Especially for flexible systems of variable configuration procedures for test, verification and correction are passed through repeatedly. In this context functional consistency means: functional entities of the system cooperate in the desired manner and operate error-free. Depending on requirements, different aspects are considered and verified separately (see Table 1).

Table 1: Important aspects of functional consistency

Aspect	Remark
Data transfer between outputs and inputs of controllers and other function blocks,	Appropriate data connection, format and sequence
Hierarchical dependencies	Interoperation of several hierarchical layers
Failure management	Defined behavior in case of malfunction
Reliability, fault tolerance, FMEA [1]	Defined behavior to avoid damage in case of failure
Real time conditions, different cycle times	Compliance with real-time conditions
Concurrency, parallel operations	Appropriate sequence and interoperation of concurrent operations
Interlocking, enabling, priorities	Conditions and procedures for turn-on/turn-off
Different modes of operation	Switching between different operational states

Given the difficulties described above, an integration of all necessary tools into one comprehensive development suite would be desirable. Although comprehensive tools do exist for confined domains (see below), this approach is not practicable for heterogeneous systems in the foreseeable future. Hence, methods and tools for the assistance of design and verification of functional structures of information processing are necessary. The aim of this paper is to present an approach for a test-framework for functional consistency, applicable to a wide range of hardware and software, and to demonstrate the application to the development of an assembly machine [1].

TEST-FRAMEWORK FOR FUNCTIONAL CONSISTENCY

In computer science, the term data consistency model is used – among other meanings – to specify consistent access to data in a distributed system. In this regard, a number of possible data consistency models are known. A system supports a given consistency model if operations on memory follow a set of specific rules. The data consistency model specifies a contract between programmer and system, wherein if the programmer follows the rules, the system guarantees that memory will be consistent and the results of memory operations will be predictable. Accordingly, a functional consistency model (for a heterogeneous system) consists of a set of functional rules specifying conditions for connection and interoperation of functional entities of the system. If a functional structure – consisting of individual functional entities – follows the defined rules and is implemented accordingly, the system is functioning failure-free and conflict-free.

In other domains where distributed hardware is used, comprehensive tools for integrated development have been implemented. For example, in process control engineering hardware and software architectures, functionalities, development tools etc. are standardized to a very large extend, see [2][3][4]. Open system standards are developed to further advance the flexibility and interchangeability of programs and hardware; see [5]. Due to this level of standardization, hardware manufacturers are able to provide integrated development tools covering the necessary range of verification. To cite another example, AUTOSAR (AUTomotive Open System Architecture) – developed by automobile manufacturers and suppliers – assists the development of networks of embedded systems in automobiles [6][7]. The AUTOSAR development framework supports the consistent distribution of functionality among several electronic control units.

The used consistency models in both domains – process control engineering and AUTOSAR – are tailored to the specific needs of supported hardware; implementation and programming are strong supported. On the other hand, these consistency models are applicable to a narrow range of specialized hardware and software only and are mainly related to data transfer and interoperability of modules. An approach for a test-framework for functional consistency applicable to a wide range of hardware and software is presented in the following.

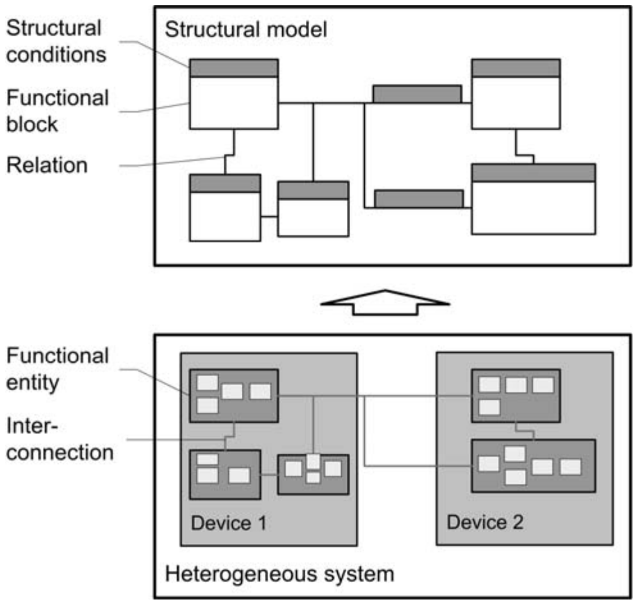

Figure 2: Structural Modeling

1. Structural Modeling: In the first step a structural model of the system is established for structural test purposes (see Figure 2). The structural model is composed of:

– Functional blocks representing the functional entities to be considered,

– Relations between functional blocks and

– Functional conditions.

Structural conditions are necessary conditions for interoperation between functional blocks to maintain functional consistency. Functional entities are – for example – controllers, pieces of control logic, arithmetic operations, data processing and supervisory control logic. The functional content of these system functions is *not* considered. Solely interfaces and relations between functional entities are taken into consideration – e.g. inputs and outputs, data transfer between entities, hierarchical relations, enabling and causal dependencies. Depending on the considered perspective, different information is incorporated. Functional blocks are connected by relations. Relations describe data transfer as well as the existence of functional dependencies between blocks. The type of dependency as well as interactions between functional blocks is described by functional conditions, e.g.: "B can be activated only if execution of A is completed". A sample controller type block with hierarchical interconnections is given in Figure 3.

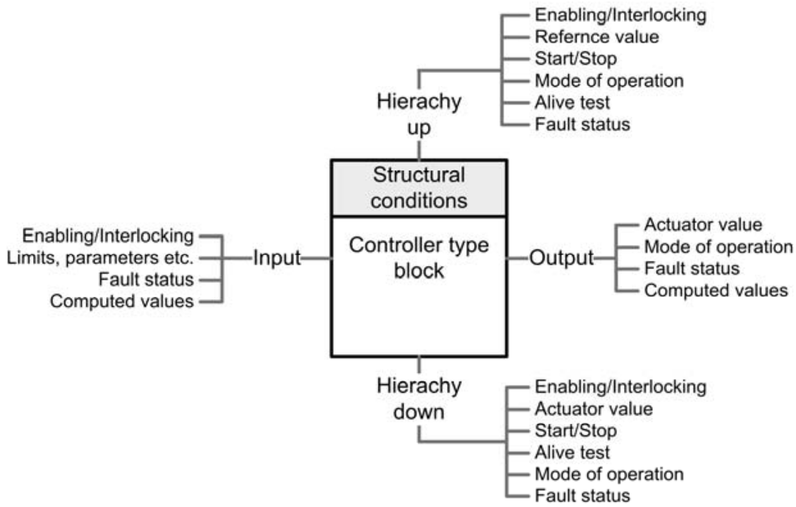

Figure 3: Sample controller type block with hierarchical interconnections

2. Consistency model and structural test: To define a consistency model, structural rule sets are defined for different aspects. Structural rules describe conditions and restrictions for consistency regarding a specific test aspect. For different test cases – e.g. failure management – several rule sets are arranged (see Figure 4). The structural rules are independent of the functional blocks, relations and functional conditions. The following rule exemplifies the possible content of structural rules: "A functional block A can be executed only if all functional blocks transferring inputs to A have been executed."

A structural verification is accomplished by testing the structural model systematically for compliance with rules of one or more rule sets.

3. Application of results: As result of the structural verification inconsistencies are indicated. In the third step the developer turns to system design again and eliminates the conflicts; results are incorporated in software coding. This procedure of test and re-design/re-programming is repeated until consistency is established.

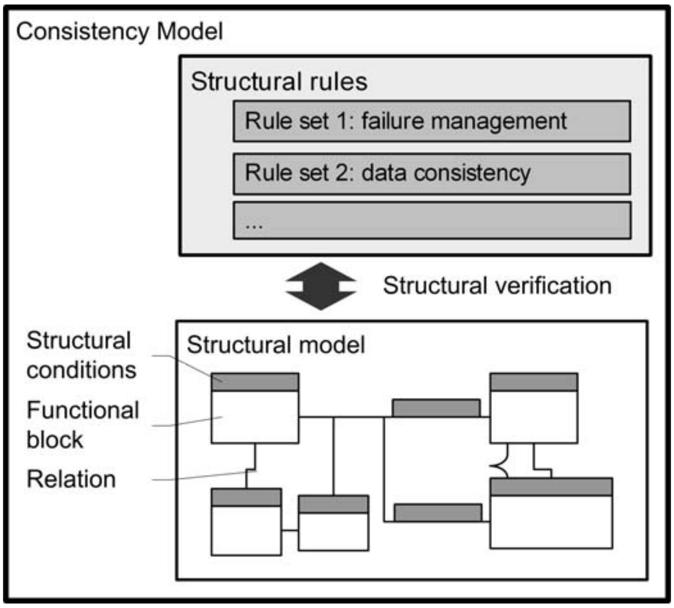

Figure 4: Consistency Model

SAMPLE APPLICATION

A section of the information processing of a servo driven assembly machine in symbolic form is given in Figure 5. It consists of equipment of heterogeneous architecture; functionality is finally coded using different tools. In the modeling of the functional structure, all functional units – e.g. controllers, interlocking logic, fault detection logic, data processing – are transformed to abstract standard blocks with specific interfaces, see Figure 6. In this process, only logical interconnections are considered, algorithms behind the blocks are not regarded. The resulting structure of connected standard blocks is used to analyze and optimize the interoperation of the blocks.

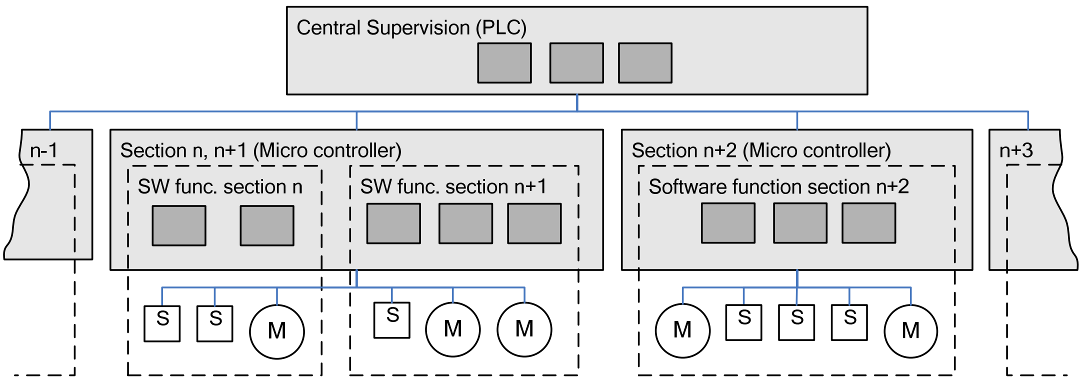

Figure 5: Servo driven assembly machine (M: Servo drive, S: Sensor)

This approach is used to eliminate fault conditions in the interaction between sections of the assembly machine [1]. The structural test based on the structural model reveals inconsistencies – e.g. incorrect sequences, exceeded time limits. Functional entities can be changed accordingly. The automatic structural test eliminates the time-consuming, error-prone manual check of the structure. At present, a semi automatic test environment is established in order to develop modelling and test methods.

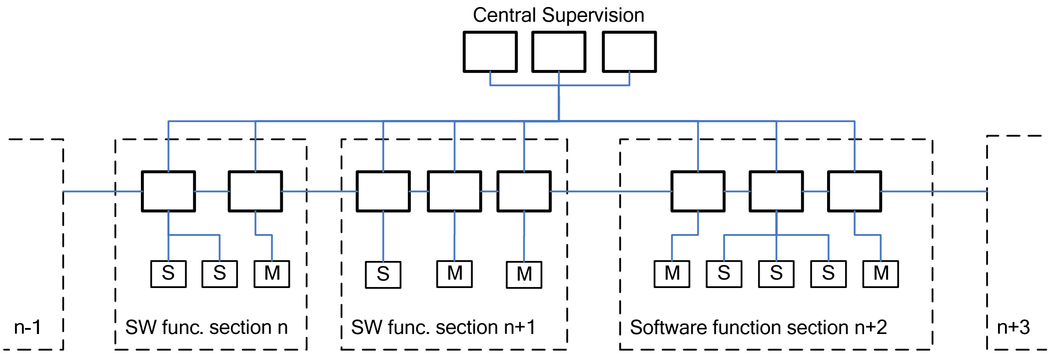

Figure 6: Model of the functional structure

CONCLUSION

Heterogeneous architectures of mechatronic systems require approaches for comprehensive structural tests during system design. Structural modeling of the system is used to create an automatic environment for device- and manufacturer-independent checking regarding different perspectives. This approach is used to check the failure management and start/stop-sequence/interlocking of an assembly machine composed of several assembly stations. Further work is necessary to extend the approach to more test perspectives and device combinations, to develop an uniform description method for functional blocks, relations, conditions and to develop a practicable software solution.

References:

[1] Lorch, M.: Machbarkeitsanalyse eines neuen Antriebskonzeptes für einen Längstaktmonatgeautomaten mit hoher Taktzahl. Diplomarbeit, Universität Karlsruhe (TH), 2007.

[2] DIN EN 61499-3. Funktionsbausteine für industrielle Leitsysteme, Teil 3: Programmiersprachen. Berlin: DIN Deutsches Institut für Normung e.V., 2006.

[3] DIN EN 61131-1: Speicherprogrammierbare Steuerungen, Teil 1: Architektur. Berlin: DIN Deutsches Institut für Normung e.V., 2003.

[4] DIN EN 61499-2. Funktionsbausteine für industrielle Leitsysteme, Teil 2: Anforderungen an Software-Werkzeuge. Berlin: DIN Deutsches Institut für Normung e.V., 2006.

[5] PLCopen, Gorinchem, NL. http://plcopen.org/

[6] Heinecke, H., Bielefeld, J., Schnelle, K.-P., Maldener, N., Fennel, H., Weis, O., Weber, Th., Ruh, J., Lundh, L., Sandén, T., Heitkämper, P., Rimkus, R., Leflour, J., Gilberg, A., Virnich, U., Voget, S., Nishikawa, K., Kajio, K., Scharnhorst, Th., Kunkel, B.: AUTOSAR – Current results and preparations for exploitation. 7th EUROFORUM Conference 'Software in the vehicle' 3-4 May 2006, Stuttgart, Germany.

[7] AUTOSAR GbR, München, http://www.autosar.org

Authors:

Dr.-Ing. Michael Kaufmann
Prof. Dr.-Ing. habil. Georg Bretthauer
Universität Karlsruhe (TH), Institut für Angewandte Informatik/Automatisierungstechnik (AIA),
Kaiserstr. 12, Bldg. 10.91, R. 203
76131 Karlsruhe, Germany
Phone: +49 (721) 608-7971
Fax: +49 (721) 608-7972
E-mail: [Michael.Kaufmann, Georg.Bretthauer]@aia.uni-karlsruhe.de